Exercise 2B-II: Diachronic Spanish name generation with a character-based language model

Ángela María Gómez Zuluaga Laura Moset Estruch

1 Introduction

We trained a character-level language model to generate names using Karpathy's makemore [1] as a base. The model was optimized to generate names in Spanish from the 21st century but can also work with datasets of names from different periods. Here, we analyze our model both on the modern dataset it was optimized for and on a dataset of older names from the 1880-1940 time period.

2 Materials and Methods

2.1 Data collection

To collect the modern names, we used an INE (Instituto Nacional de Estadística, Spain) database [2] containing all names in Spain with a frequency of at least 20 individuals as of 01/01/2022. We began by extracting the first 1000 names for each sex. However, we quickly realized that, due to Spain's ageing population, many of these names belonged to people with an average age of 70 or older, which would place them close to the upper limit of our older database.

To address this, we excluded names with an average age of 60 or older. We then selected the top 500 names for each sex that fit this criterion, so the names would not be too rare (the least common name in our database has a frequency of ≈2900 people). These steps helped ensure the database was modern compared to our 1880-1940 dataset.

For the 1880-1940 dataset, we used a corpus of Spanish novels published between 1880 and 1940, compiled by Calvo Tello [3]. We used spaCy's entity recognizer (ner) with the `es_core_news_lg` trained pipeline, selecting entities where the `ent.label_=="PER"`. After this step, we observed that many of the extracted entities were not names but strings of words. To address this, we applied a filter to exclude words found in the *Diccionario de la Lengua Española* (DLE), using a list extracted by Dueñas Lerín [4]. Considering that this would exclude names that are also words in the Spanish lexicon (e.g., "Rosa"), before this step, we prefiltered the dictionary with our dataset of modern names to remove words that also function as names. With this, we reduced the number of names that would be erroneously eliminated from the old dataset.

After these steps, we obtained a final dataset of ≈5000 unique names, which we manually cleaned by removing surnames, full names, irregular characters, archaic verb inflections, and honorifics. After returning the set of unique names for a final time, we were left with a dataset of 808 names in total.

2.2 Training

Great effort was spent cleaning the datasets to ensure better results once training started. We not only experimented with different model hyperparameters but also different dataset versions, as illustrated above. For the training process, the datasets were divided into three sections: training (80%), validation (10%) and test (10%), following Karpathy's code [1]. We experimented with many combinations of hyperparameters by changing the number of neurons (200, 150, 100), the minibatch size (16, 32, 64), the learning rate (0.7, 0.5, 0.1, 0.05, 0.01), and the number of steps (200,000, 150,000, and 100,000). The lowest loss and overfitting we obtained can be observed in the final configuration shown in Table 1. Note that the model optimization was performed for the modern names dataset; we did not attempt to optimize for the 1880-1940 dataset.

2.3 Evaluation

We quantitatively assessed the model's improvement based on the reduction of the cross-entropy loss. We found this metric appropriate, as calculating probability distributions at a per-character level can be considered a classification task (for which cross-entropy is well-suited). This reduction did not only target

the training loss; we also aimed for a configuration where the validation and test losses were closer to the training loss to decrease overfitting.

We also performed a qualitative evaluation in which each author provided a binary score ("1" for yes or "0" for no) for 50 names generated from modern and older datasets. This assessment was based on the subjective judgement of whether the generated names looked like real possible names in the language.

3 Results

Ultimately, we decided to keep the configuration labelled as "Final" in Table 1. Although the losses vary slightly each time the code is run, they mostly stay within the results shown below due to the manual seed (2147483647) we implemented for the Torch generator. Once trained on the 1880-1940 dataset, the model performs slightly worse on average, as shown in the table below.

The most interesting finding from our experiments was that, although the training loss slightly increased for the final version compared to the baseline, we minimized the test loss to a value closer to that of the training loss. In principle, this should mean that we successfully reduced the chances of the model overfitting, although this issue probably still needs solving as the values are still dissimilar.

Config	Parameters Used	Modern Dataset	Old Dataset
Initial	neurons: 200 minibatch size: 32 learning rate: 0.1 / 0.01 steps: 200000 / 100000	Tr. loss: ≈1.05 Val. loss: ≈3.36 Test loss: ≈3.08	
Final	neurons: 100 minibatch size: 32 learning rate: 0.7 / 0.05 steps: 150000 / 75000	Tr. loss: ≈1.22 Val. loss: ≈2.41 Test loss: ≈2.26	Tr. loss: ≈1.34 Val. loss: ≈2.79 Test loss: ≈2.83

Table 1: Cross-entropy loss. This table compares the model's initial and final loss results. The initial configuration is the baseline obtained from Karpathy's makemore. *Learning rate:* the value after the slash is the rate implemented once the decay threshold is reached. *Steps:* the value after the slash is the point where the learning rate decay is implemented.

The findings from the qualitative evaluation indicate that the mean percentage of name-looking generations (according to the authors) was 73% for the modern dataset and 62% for the older one. Additionally, while not easily quantifiable, the authors noticed a difference in the 'aspect' of the names produced by the two datasets. Some examples from the modern dataset were: iñakal, soray, cristere —the "k", or the final "y" and "e" may give them a more modern feeling—. Further, names generated from the older dataset look like: amelito, clautistiana, victora (for a sample of the generated names, see here: https://osf.io/786jp). Thus, although the model could improved to fit the older dataset, the existing model still produced decent results.

Moreover, we also analyzed how many of the names generated were already in the original dataset. With this, we aimed to see if the model was merely copying the training data or, on the contrary, not extracting from the data at all. Although the number of names that matched the datasets varied on each model execution, the model trained on the modern dataset consistently generated more matching names than the older one. This indicates that our model learns more from the modern dataset than the older dataset, which would be in accordance with the final loss values for both datasets. However, the older dataset still returns viable names.

As a final note, there is an essential constraint to our training methodology. Our current knowledge limits us to manual tweaking to determine the most suitable hyperparameters, instead of using more refined methods. However, we improved the model from the baseline despite these constraints. Regarding the data, for future explorations, there could be value in attempting a weighted dataset, giving greater weight to names based on their frequency in the given time period for a more refined name generation. In addition, our datasets were relatively small for what language models are usually trained on, but we still obtained satisfying results.

4 Code

The code for this project and all relevant files are publicly hosted on OSF (https://osf.io/pvrxj/).

5 List of contributions

Both authors contributed to every step of the process: Conceptualization, Software, Analysis, Methodology, Investigation, Data Curation, and Writing.

6 References

[1] Karpathy, A. (2022). makemore. Retrieved from https://github.com/karpathy/makemore. Last accessed on March 1, 2024.

[2] Instituto Nacional de Estadística. (2023). Todos los nombres con frecuencia igual o mayor a 20 personas. Last accessed on March 1, 2024. https://www.ine.es/uc/nDER2igi

[3] José Calvo Tello (Comp.) (2017). Corpus de novelas de la Edad de Plata Würzburg: CLiGS, 2017. Last accessed on February 29, 2024. https://github.com/cligs/textbox/tree/master/spanish/novela-espanola.

[4] Jorge Dueñas Lerín (2022). List of all Spanish words in RAE. Last accessed on February 29, 2024. https://github.com/JorgeDuenasLerin/diccionario-espanol-txt/blob/master/0_palabras_todas.txt.